

Search Forms
Search Results

Freeform Search

Help

User Searches

Preferences

Logout

Database:

US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Term:

Display: **Documents in Display Format:** **Starting with Number**

Generate: ☐ Hit List ☒ Hit Count ☐ Side by Side ☐ Image

Search

Clear

Interrupt

Search History

DATE: Monday, September 06, 2004 [Printable Copy](#) [Create Case](#)

<u>Set</u> <u>Name</u>	<u>Query</u>	<u>Hit</u> <u>Count</u>	<u>Set</u> <u>Name</u> result set
side by side			
<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>			
<u>L26</u>	L25 and (terminal or computer)	76	<u>L26</u>
<u>L25</u>	L24 and (customer or user)	76	<u>L25</u>
<u>L24</u>	L22 and (logon or signon)	76	<u>L24</u>
<u>L23</u>	L22 and (customer or user) near (terminal or computer) near (logon or signon)	0	<u>L23</u>
<u>L22</u>	(atm near machine or automated near bank\$ near system)	5861	<u>L22</u>
<u>L21</u>	382.clas.	42989	<u>L21</u>
<u>L20</u>	902.clas.	2032	<u>L20</u>
<u>L19</u>	705.clas.	28812	<u>L19</u>
<u>L18</u>	116 and (check near clear\$ near system or ach near system or check with clearinghouse or check with clearing with house)	5	<u>L18</u>
<u>L17</u>	705/39	1636	<u>L17</u>
<u>L16</u>	automated near bank\$ near system	125	<u>L16</u>
<u>L15</u>	902/8	252	<u>L15</u>
<u>L14</u>	902/6	141	<u>L14</u>

L13 902/36
L12 902/21
L11 382/175
L10 382/140
L9 382/139
L8 382/138
L7 382/136
L6 382/135
L5 902/18
L4 382/139
L3 382/137
L2 5751842.uref.
L1 5751842.pn.

47 L13
77 L12
533 L11
227 L10
239 L9
211 L8
64 L7
723 L6
115 L5
239 L4
464 L3
20 L2
2 L1

END OF SEARCH HISTORY

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

Generate Collection

Print

L26: Entry 62 of 76

File: USPT

Aug 3, 1999

DOCUMENT-IDENTIFIER: US 5933816 A

**** See image for Certificate of Correction ****

TITLE: System and method for delivering financial services

Abstract Text (1):

A delivery system and method allow a financial institution to provide financial services to a plurality of remote devices, such as personal computers, personal data assistants, and screen phones. In addition to providing services to these remote devices, the system and method provide services to automatic teller machines (ATMs), external service providers, and internally within the financial institution to staff terminals and to the individual branches of the financial institution. The delivery of financial services is not limited to any particular network but rather may be provided through dial-in access, Internet access, on-line service provider access, or other types of delivery networks. The system is comprised of a set of reusable global components which are modular and are organized into services sets. By separating the components of the system into independent components, the system and method can be developed and tested on a component level rather than the entire system level, thereby substantially reducing the development and maintenance cycle time. The system and method operate in sessions and, for instance, employ a dialog component for gathering information from a customer, a rule broker component for providing answers to the various legal and regulatory rules in a particular country, a language man component for selecting appropriate language, a transaction executor component for performing transactions, and a presentation manager component for formatting outputs to the customer. The system and method provide state-of-the art interfaces with interface components and support legacy applications with legacy app bridge components.

Brief Summary Text (6):

Banks and other institutions that provide financial services are facing increased amounts of competition and are being pressured to provide a greater diversity of services to their customers. Not too long ago when customers traveled to the bank to make all of their transactions, the bank could focus on the customer-bank teller interaction to improve the quality of services. The bank could improve the quality of service by staffing a larger number of tellers at peak times and by offering drive through services. Banks developed internal computer systems and provided their tellers with staff terminals so that the bank tellers could access the books of the bank when they were entering customer transactions.

Brief Summary Text (7):

This simplified model of banking, while still in existence, has been greatly expanded. In addition to the bank tellers, banks provide Automated Teller Machines (ATMs) so that customers can perform transactions at literally any hour of the day. The locations of the ATMs are not limited simply at the bank's branch locations but can be found, for instance, in shopping malls, airports, grocery stores, hotels, and building lobbies. Since the ATMs must access the books of the banks to allow customers to perform their transactions, banks had to provide an interface between the ATMs and the bank's internal computer system that allows the ATMs limited but secure access to the bank's books.

Brief Summary Text (8):

The model for providing financial services has been expanded even further to enable home banking. With home banking, a customer can access his or her personal account and perform transactions, such as bill paying, money transfers, etc., in the convenience of one's home through the customer's personal computer. To enable home banking, banks had to provide an interface between the bank's internal computer system and the customer's personal computers to allow limited and secure access to the bank's books. Due to the differences between ATMs and personal computers, the interface for the personal computers is typically separate and distinct from the interface provided for the ATMs.

Search Forms

Brief Summary Text (9):

One difficulty facing banks is that they must provide a first interface between the bank's internal computer system and the staff terminals, a second interface between the internal computer system and the ATMs, and a third interface between the internal computer system and the personal computers. Each of these interfaces or platforms adds complexity to the bank's overall computer system and each competes with the other for access to the bank's books. This added complexity is significant since the amount of resources that the bank must devote toward maintaining its computer system is increased due to these three separate platforms.

Brief Summary Text (10):

The complexity of a bank's computer system will only grow as banks continue to provide more and more services. One area of service that banks are already beginning to explore is the granting of access to the bank's books by devices other than personal computers, such as screen phones or personal data assistants (PDAs). Another area of service that banks are contemplating is the granting of access to the bank's books through the Internet, such as through an Internet service provider (ISP) or other external service provider (ESP). These additional remote devices and the connection to the ISP would further complicate the bank's internal computer system and would require the bank to devote more resources in maintaining and upgrading its computer system.

Brief Summary Text (11):

In addition to additional channels of access to the bank's books, banks have been expanding the types of services that can be accessed by a remote device. In addition to traditional checking and savings accounts transactions, banks are also enabling the paying of bills, the buying and selling stocks, the quoting of stocks, as well as other types of services through the ATMs, personal computers, or other remote devices. Each expansion into another type of service requires a significant amount of modification to the software and possibly hardware in the bank's internal computing system. These additional services, although necessary to remain competitive, require a considerable amount of work on the bank's computer system.

Brief Summary Text (12):

The complexity in offering financial services will also be compounded as more and more services are being provided in an international market. As individual national markets in the world continue to merge together, both businesses and individual customers will have an increased need and desire to access their account information from another country. The platform or platforms needed to interface with the banking systems of other countries will further tax the ability of a bank to maintain its computer system and its books.

Brief Summary Text (13):

Another difficulty facing a bank entering another country is that the bank must, in effect, create a new computer system for each country that it enters. Each country has its own unique regulatory and legal environment which dictates the manner in which financial services must be performed. The bank cannot simply duplicate a computer system operating in another country but rather must specially tailor each computer system to the regulatory and legal environment of that country. This extra amount of effort required to shape a computer system according to the rules and

laws of its home country consumes more of the bank's valuable resources.

Brief Summary Text (14):

The creation of computer systems for banks in other countries is complicated by the difference in languages. Because of a language difference, the interface between the computer system and the customer, such as through a graphical user interface (GUI), will vary according to the national language or languages of a particular country. These differences in how customers interface with the bank's computing system are not limited simply to words and different alphabets but also encompass manifestations of language due to differences in culture or norms of a particular country. These manifestations of language, for instance, may dictate the selection of certain colors in a GUI, the use of a particular set of symbols, and the selection of certain audio indicators, such as beeps or other tones.

Brief Summary Text (15):

In view of a desire to allow international access to the bank's books, each computer system should also be able to receive and send communications from the other computer systems, which possibly may be in numerous languages. For instance, the bank's books which are stored in the United States may need to be accessed by a newly installed banking system in Thailand. Before this access is possible, however, the banking system in the United States may need to be modified to recognize the bank's computer system in Thailand. Consequently, with the introduction of each computer system in one country, the computer systems in all of the other countries may need to be modified accordingly.

Brief Summary Text (16):

As discussed above, banks are increasingly providing new types of access to the bank's books and are providing new services in an ever-increasing number of countries. All of these changes in services and access to services are accomplished by rewriting the application which governs all operations of the bank's internal computer system. Due to the extremely sensitive nature of the data in the bank's books and the need for total accuracy, any modification to this application must undergo an inordinate amount of testing. Since the application covers all aspects of the system, any modification to one part of the application which may cover only a minor aspect of the system can potentially have an effect on any other part of the system. The testing required for any modification, even for just a minor upgrade, must therefore be performed on the entire system.

Brief Summary Text (17):

Thus, a need exists for a computer system or method that has a reduced amount of complexity yet offers access to various remote devices and enables the expansion of access to new types of devices. A need also exists for a computer system or method that can offer new or modified services more easily with less testing. A need also exists for a computer system or method that can more easily accommodate the legal and regulatory environment of a host country and which can more easily interconnect and communicate with the systems in other countries.

Brief Summary Text (19):

The invention, in a preferred embodiment, is a system and method for delivering financial services to a remote device. Through the remote device, a customer or employee of a financial institution can select a mini-app dialog component to perform a function. Preferably, each function that may be performed is represented by a separate mini-app dialog component. Upon selection of a function, the mini-app dialog component collects information needed to perform the requested function and instantiates a transaction executor component to carry out the function. The remote device may comprise any type of device, such as a personal computer, screen phone, ATM, personal data assistant, or an internal staff terminal. The remote device may access the system in a variety of ways, such as through an external service provider, through the Internet, or through dial-up access. Thus, the system provides a single base for interfacing with all types of remote devices.

Brief Summary Text (20):

In generating graphic interfaces, the system and method preferably separate content from format to accommodate variations in the remote devices. The system includes a presentation manager which maps messages from a canonical representation into the format desired for a particular remote device. The content of the messages is regulated through a language man component. In response to a request for a named phrase, the language man component provides the phrase in the language and the content specific for that customer and that remote device. As a result, the system and method can provide state of the art user interfaces, can provide interfaces consistent for a financial institution, and can allow a customer to custom design a user interface.

Brief Summary Text (21):

The system and method operate in sessions with a session bubble instantiated for each session with a remote device. After receiving an initial contact with a remote device, a session controller instantiates a session component to manage resources for the session bubble. The session component, in turn, instantiates a number of components for the session, such as a welcome mat component, front door man component, rule broker component, and acquirier component. The welcome mat component sends a login message to the remote device and instantiates a profile transaction executor component to authenticate a customer. A navigation shell component notifies the remote device of the list of available functions, such as cash withdrawal or bill payment, and instantiates a mini-app dialog component based on the function selected through the remote device. To coordinate communications with the plural sessions that may occur simultaneously, a touch point interface component routes incoming messages from remote devices to the appropriate session bubbles and a back door man component coordinates messaging between the various sessions and an external service provider.

Brief Summary Text (26):

It is another object of the present invention to provide a delivery system and method that provide and can easily maintain state of the art user interfaces.

Drawing Description Text (6):

FIGS. 5A to 5D are flow charts depicting operations of the delivery system in authenticating a customer.

Drawing Description Text (8):

FIGS. 7A and 7B are flow charts depicting operations of the delivery system in the selection of a mini-application by a customer.

Detailed Description Text (2):

Reference will now be made in detail to the preferred embodiment of the invention, an example of which is illustrated in the accompanying drawings. The invention is described with reference to a system 10 for use by a bank, although the system 10 may be employed by any type of institution offering financial services. The financial system 10 includes a delivery system 12 for providing financial services to a variety of remote devices. These remote devices include a screen phone 14, an automated teller machine (ATM) 16, such as Citibank's CAT/CASST terminals, a personal computer 18, or a personal data assistant (PDA) 20. The remote devices can practically be any type of device and can be installed with any suitable software for communicating with the delivery system 12, such as a standard web browser or any other third party software product. The remote devices that the delivery system 12 can provide financial services to is therefore not limited to any particular class or type of remote device but instead may include any future device or system. Further, the delivery system 12 provides services not only to customers of a financial institution but may also provide services internally to the institution, such as at staff terminals 26.

Detailed Description Text (3):

The delivery system 12, furthermore, provides financial services over a plurality of different delivery networks. As an example, the delivery system 12 may deliver financial services to the screen phone 14, personal computer 18, or PDA 20 via dial-up access or through an application server, such as the home services delivery system (HSDS), which is disclosed in U.S. Pat. No. 5,485,370 to Moss et al. and which is hereby incorporated by reference. Alternatively, the delivery system 12 may provide financial services to remote devices 24 through an Internet Service Provider (ISP) 22 or an on-line service provider 22, such as through the Internet or World Wide Web. The delivery system 12 advantageously is able to provide financial services over a variety of communication paths, such as the Internet, a land-line telephone network, a cellular network, or a cable network, and can be easily modified to operate over new transmission paths or new transmission protocols.

Detailed Description Text (5):

With reference to FIG. 2, a delivery system 12 according to a preferred embodiment of the invention comprises plural sets of service components. These sets of service components include a touch point and display set 30, a touch point interface services set 40, and a touch point services set 50. In general, the touch point and display set 30 provides actual customer display and input facility and the touch point interface services set 40 provides an interface to the touch point services set 50. The touch point services set 50 provides presentation mapping and front door security for the delivery system 12. The delivery system 12 also includes a peripheral device services set 60 providing peripheral device interface and management services. A system services set 70 provides logging, event brokering, service registry and crypto services and a dialog services set 80 provides welcoming, navigation shell and application specific dialogs. A transaction services set 90 provides transaction coordination and ESP message formatting and an external service provider interface services set 100 provides message sequencing and ESP interface protocols. A customer services set 110 provides customer identification, relationship, account, acquirer, and issuer services and a business services set 120 provides rule brokering and language, services. A session services set 130 provides session start up and session and delivery vehicle context.

Detailed Description Text (7):

The touch point and display set 30 provides the actual customer display and input facility on the remote device. The touch point and display set 30 includes a touch point and display component 31 that displays pages on the remote device screen and sends customer inputs to the delivery system 12. The touch point and display component 31 is responsible for managing the link/session level protocols with an application server on the remote device. The touch point and display component 31 also decodes the server interface protocol and outputs a page to the local screen of the remote device. The touch point and display component 31 acquires customer inputs, including choice selections and forms input, encodes the input in the server interface protocol, and sends the customer input to the touch point interface set 40. For Internet sessions with the delivery system 12, the touch point and display component 31 preferably comprises a web browser that handles protocols such as TCP/IP, HTTPS, and, less preferably, FTP.

Detailed Description Text (12):

The presentation manager component 52 is responsible for mapping a canonical representation of information on pages into a specific style layout in a device specific presentation format. Thus, the same application can have different presentation styles on different device types. For instance, the same application may have different presentation styles depending on whether the application is displayed on a personal computer 18, a PDA 20, a screen phone 14, a CAT 16, a third party kiosk terminal, or another type of remote device. The style templates can be customized by region to support local cultural differences in areas such as color schemes, graphics, icons, and font sizes. The presentation manager component 52

maps tagged phrases and data from the application into specific fields of a particular page template referenced by the application. A template controls the layout and representation of frames on a page, multi-media elements, choice and data fields, and input forms on the page for a specific style and device type. The presentation manager component 52 also encodes the resulting page in the device specific format for the particular remote device and sends the page to the front door man component 51. The presentation manager component 52 also receives incoming messages from the remote device, converts choice information and form fields from the device specific format to a tagged canonical representation, and routes the representation to the appropriate component within the dialog services set 80. The presentation manager component 52 uses delivery system specific templates to enforce consistent layout styles across pages having similar choices, data fields, and forms. A template can be the superset of all possible objects on a page since the presentation manager component 52 can "drop out" fields and choices which are not associated with any data. Reference is made to a related application Ser. No. 08/741,121, "Method and System for Automatically Harmonizing Access to a Software Application Program Via Different Access Devices," filed Oct. 30, 1996 which is incorporated herein by reference.

Detailed Description Text (16):

The peripheral device manager component 62 manages the components that interface with the connected peripheral devices. The peripheral device manager component 62 loads the peripheral device handler component 61 for the connected devices during startup and initializes the peripheral device handler component 61 during startup. The peripheral device manager 62 notifies interested parties of changes in peripheral device availability, finalizes the peripheral device handler component 61 during shut down, and unloads the peripheral device handler component 61 during shut down. In addition to its responsibilities for device management, the peripheral device manager component 62 also provides application services. For instance, the peripheral device manager component 62 coordinates usage of the peripheral devices by customers versus diagnostics and serializes application requests to each peripheral device. The peripheral device manager component 62 also routes each application request to the appropriate peripheral device handler component 61 and reports status of all connected peripheral devices upon request.

Detailed Description Text (25):

The crypto man component 75 performs cryptographic functions necessary to handle security. The crypto man component 75 manages secret keys associated with external service providers and performs authentication of public key certificates. The crypto man component 75 holds security keys for each external service provider, which may be multilevel keys for each external service provider. Further, the keys may be shared secret or private key associated with a public key. The crypto man component 75 also updates keys and uses keys to generate message MAC and encrypt message. The crypto man component 75 also encrypts and re-encrypts customer PIN/TPIN.

Detailed Description Text (28):

The test manager component 78 manages the testing and tracing of components in the system 12. The test manager component 78 collects information from the various components in the system 12 by wiring itself into them during component creation. Then, the components that have been wired for test report method entries and exits to the test manager component 78 during their operation. The configuration of which components are under test or trace can be driven by scripts or by an on-line test management user interface. The test manager component 78 records information reported by the components under test in a log or it can report the test results to the tester through the test management user interface. The test manager component 78 therefore knows which components are under trace and test and wires new components for tracing and testing.

Detailed Description Text (30):

The dialog services set 80 is responsible for the semantic content and interaction with the customer and for initiating transactions on the customer's behalf. The dialog services set 80 includes a welcome mat component 81, at least one navigation shell component 82, at least one mini-app dialog component 83, and at least one legacy app bridge component 84. Although the navigation shell component 82, the mini-app dialog component 83 and the legacy app bridge component 84 have been shown as single components, each of these components may comprise a plurality of components.

Detailed Description Text (31):

The welcome mat component 81 outputs the initial welcome page to the customer and collects customer identity and preference information. After determining the issuer of the customer ID and possibly authenticating the customer, the welcome mat component 81 instantiates several customer services objects to hold information about the customer and then starts a navigation shell component 82 which carries out the next level of dialog with the customer. The welcome mat component 81 establishes connection sessions with a back door man component 101 in the ESP interface services set 100 as needed by a session. The welcome mat component 81 also acquires devices needed by the session and creates a scam transaction executor to handle unsolicited scam events from a host. The welcome mat component 81 presents an out of service or welcome page, enables a card reader, and waits for card read events. If the card event is an administration card, the welcome mat component 81 instantiates an administrative welcome mat component. The welcome mat component 81 collects various information from the customer including language choice and other preferences, such as navigation style. The welcome mat component 81 also collects customer ID information, such as CIN/PIN and public key certificate, in a manner consistent with the customer remote device and mode of access, such as dial-in or Internet. The welcome mat component 81 handles retries if errors occur on customer identity input, for instance by re-reading a card, and asks customer ID component 111 for issuer. The welcome mat component 81 instantiates a profile transaction executor component 91 to authenticate the customer and get the customer's relationships or customer profile. This process typically involves interactions with the issuer external service provider, but may alternatively be performed locally based on information in a SmartCard. The transaction executor component 91 instantiated by the welcome mat component 81 will instantiate the following customer service components: customer ID component 111, customer relationship component 113, account component 115, and issuer component 112. The welcome mat component 81 will also initialize legacy app bridge components 84, and start a navigation shell component 82 based on delivery capabilities, acquirer rules, and customer preferences.

Detailed Description Text (32):

The welcome mat component 81 may rely on separate micro-app dialog subcomponents to do some parts of the dialog that may be common to several business functions or which may vary depending on the remote device peripherals. For instance, the welcome mat component 81 may rely on a hello screen micro app, a language select micro app, and a get PIN customer data micro app.

Detailed Description Text (33):

The welcome mat component 81 may do four things for customer authentication based on acquirer rules and the type of customer ID, such as public key certificate, ATM card, credit card, on-us, or off-us. The welcome mat component 81 may provide immediate local authentication using public key certificates or may provide immediate authentication with the issuer, waiting for a response. The welcome mat component 81 may also provide background authentication with the issuer while going on to the navigation shell component 82 or may defer authentication to the first transaction. With deferred authentication, the welcome mat component 81 may need to instantiate a default customer relationship component 113 and a default set of product types, such as checking, savings, or credit card. If a rule broker component 121 does not have a registered issuer for the card/CIN prefix number, a

customer ID component 111 is instantiated and marked invalid, further authentication of the customer is skipped, and a navigation shell component 82 for invalid customers is started. Invalid customers may still be allowed to use certain information only in mini-app dialogs.

Detailed Description Text (34):

The navigation shell component 82 informs the customer of the range of mini-apps that are available and provides top level navigation across these applications. The navigation shell component 82 assigns a frame space within which a mini-app runs. To support complex grouping of functions or a variety of navigation styles, the navigation shell component 82 may contain shells within shells. The navigation shell components 82 available for selection by a customer include linear, which guides customers through detailed question and answer steps; nonlinear broad branching, such as pull-down menus; preferred, such as customer specified short cuts; or query, which may include a search engine or natural language searching capabilities. The navigation shell component 82 obtains lists of possible services available from services registry component 74, checks rules to see what services are actually available in the current system context, and makes the customer aware of the range of mini-apps available. The range of mini-apps available will be based on the customer's relationship, the issuer/acquirer rules, and the set of dynamically registered mini-apps. The mini-apps may be organized and identified by the navigation shell component 82 with names, icons, or any other type of representation. The navigation shell component 82 instantiates additional navigation shell components 82 as necessary and instantiates mini-app dialog component 83 as requested by the customer. The navigation shell component 82 supports switching between concurrently active mini-app dialogs and, at the end of a session, instantiates and calls end of session mini-app. The delivery system 12 preferably supports the customer leaving a mini-app to enter the navigation shell component 82 and to start another mini-app, while leaving the former mini-app suspended in its current context state. The customer can later exit from the new mini-app and go back to the former mini-app or can switch between multiple concurrently active mini-apps. In an environment where the screen has imbedded frames, a main navigation shell component 82 may, for example, invoke one or more sub shell components 82 to control individual frames.

Detailed Description Text (35):

The mini-app dialog component 83 manages the dialog with a customer for a specific business function in a specific dialog style. The mini-app dialog component 83, for instance, may manage the business functions of transferring funds or bill payment in the styles of question and answer or forms. The mini-app dialog component 83 presents information and choices to the customer and collects and validates customer inputs. The mini-app dialog component 83 is responsible for the content of information on pages and the flow of the customer interaction, but preferably not the style and layout of the presentation. The mini-app dialog component 83 may comprise several different mini-app dialog components 83 with different dialog styles for the same business function. The mini-app dialog components 83 may support different modes of the customer entering information, such as guiding the customer through detailed question and answer steps or forms with multiple input fields. After collecting the necessary customer inputs for a particular business function, the mini-app dialog component 83 uses a transaction executor component 91 to carry out the function by doing transactions with external service providers and operating peripheral devices, such as a cash dispenser or depositor. The mini-app dialog component 83 implements the customer-visible control flow for a particular function in a specific dialog style. The flow may be tailored based on the customer relationship and on various countries/business rules. The mini-app dialog component 83 uses a language man component 122 within the business services set 120 to do translation of phrases into target languages for display or print. The mini-app dialog component 83 assembles phrases and formatted data into pages, for display or print, with each page constructed in a canonical format by setting properties of named objects within named templates. The mini-app dialog component 83 sends pages

to the presentation manager component 52 which handles the final style and layout for the specific remote device. The mini-app dialog component 83 collects customer inputs and validates customer inputs using business rules. Validation, for instance, includes basic field validations as well as cross-field validations. The mini-app dialog component 83 instantiates and calls transaction executor components 91 to do transactions with external service providers and also operates remote devices, such as a cash dispenser or a depositor, needed by the business function. The mini-app dialog component 83 queues transaction data for printed record and increments transaction counters in the instrumentation component 76. A mini-app dialog component 83 may, for instance, use separate mini-app dialog subcomponents 83 to do some parts of the dialog that may be common to several business functions, such as PIN entry, account resolution, and entering currency amount.

Detailed Description Text (36):

The legacy app bridge component 84 is a bridge that enables a legacy application set to operate in the delivery system 12. The legacy app bridge component 84 translates data between customer and business services objects in the delivery system 12 in the form that data is stored in the legacy applications. A different legacy app bridge component 84 may exist for each type of legacy application set, such as USCAT, AsiaCAT, LatinCAT, and EuroCAT. On entrance to a legacy application, the legacy app bridge component 84 obtains data from the session services set 130 and customer services set 110 and translates the data into the global data structures needed by the legacy application. On exit from a legacy application, the legacy app bridge component 84 takes modified data from the legacy structures and puts the data back to the customer services set 110 within the delivery system 12. The legacy app bridge component 84 translates legacy pages into the canonical page structures needed by the presentation manager component 52 and interfaces with the back door man component 101 to send messages to external service providers. The legacy app bridge component 84 also interfaces with the logger for logging errors and transactions. During initialization of the legacy app bridge component 84, the rule broker component 121 and various rule authorities, primarily acquirer and issuer, may need to be interrogated to obtain data needed to populate static tables used by the legacy applications for processing rules. Depending upon the extent of migration, the legacy app bridge component 84 may have several different relationships between it and the navigation shell component 82. For instance, the navigation shell component 82 may provide the top level navigation across the new mini-app dialog component 83 as well as the individual legacy app bridge component 84. For some card types and issuers, the navigation shell component 82 may be faceless and all business functionality is provided by the legacy apps. In this alternative, top level navigation may be provided within the legacy applications. For CAT applications, one of a pool of CAT/TAFE runtimes will be assigned to a session at start-up. The legacy applications will be assigned a frame space within which the navigation shell component 82 "plays" its applications. Individual CAT level 3 functions will be individually registered and exposed. The navigation shell component 82 supports exposing CAT level 3 functions without the need to traverse the existing level 2 menu structure.

Detailed Description Text (39):

Each transaction executor component 91 performs a particular business function, such as cash withdrawal, by doing transactions with external service providers. The transaction executor component 91 validates properties of data obtained from mini-app dialog components 83 to determine whether the required information needed to do the transaction exists. If the data is missing, the transaction executor component 91 immediately returns an error. The transaction executor component 91 collects additional information needed to do the transaction from other objects, such as customer ID component 111, acquirer component 114, issuer component 112, or account component 115. The transaction executor component 91 formats messages to be sent to external service providers and orchestrates complex transactions by sending messages to multiple service providers, serially or concurrently, as necessary. The transaction executor component 91 also parses response messages and returns

information as properties of a transaction object and recovers from external service provider transaction failures. The transaction executor component 91 may also reverse transactions during a recovery. The transaction executor component 91 calls system logger component 72 to record an audit trail of transactions.

Detailed Description Text (44):

I. Customer Services Set 110

Detailed Description Text (45):

The customer services set 110 provides a category of services that includes all information specific to the customer who initiates a session. All information related to identifying the customer, the issuing business of the customer, the customer's profile, and all the customer's accounts are the component objects included within this category of services. The customer services set 110 includes a customer ID component 111, an issuer component 112, a customer relationship component 113, an acquirer component 114, and an account component 115.

Detailed Description Text (46):

The customer ID component 111 contains information and answers questions about a customer's identity and associated information. The customer ID component 111 supports query of customer ID and card information, supports update of customer ID and card information, and identifies card issuer. The customer ID component 111 knows the customer primary ID including the CIN, encrypted PIN/TPIN, and public key certificate. The customer ID component 111 also knows the status and profile action code indicating ID validity: valid, invalid, or unknown. The customer ID component 111 has card information, if a card was used, including the type of card, such as ATM, credit card, SmartCard, and tracks present and track data. The customer ID component 111 knows the tier of service a card supports, the advisory message text to be displayed, the primary relationship type code, and the deposit only flag. The customer ID component 111 has links to account list, an issuer list, and a customer relationship list. The customer ID component 111 may also store the name of a customer, mail address of customer, E-mail address of customer, and phone numbers of the customer and provide this information to the customer or external service provider 22 so that this information does not have to be requested more than once.

Detailed Description Text (47):

The issuer component 112 represents the issuing business for the customer-ID information that was used to start a session. The issuer component 112 is the rule authority for all general, issuer related, non mini-app specific business rules. The issuer component 112 supports query of issuer information and supports answering questions about general issuer business rules. The issuer component 112 has information about the issuer of customer's identity, for instance, business code, financial institution identifier, and issuer type, such as bank card, credit card, or other third party card. The issuer component 112 knows the PIN length supported and the issuer country and ISO currency code for the issuer default currency. The issuer component 112 has a list of customer relationships for the issuer and a list of accounts for the issuer. The issuer component 112 also knows the products and services supported and the transaction and product limits. The issuer component 112 is informed of the issuer's presentation rules, such as data, format, and account number masking, and the issuer's local rules, such as collect call support, currency, and product names. The issuer component 112 also knows the issuer's servicer-ESP communication rules, for instance, profile message support, the languages supported, and the navigation schemes supported. The issuer component 112 knows when or how to authenticate customer, such as by local validation of public key certificate, immediate to issuer, background to issuer, or delayed to first transaction.

Detailed Description Text (48):

The customer relationship component 113 contains information and can answer questions about a customer's relationship. The information contained within the

customer relationship component 113 includes the accounts and products owned by the customer, customer type, preferences and privileges. The customer relationship component 113 supports query of customer relationship information and supports update of customer relationship information. The customer relationship component 113 knows the owner of the customer relationship or issuer, the customer relationship ID, the customer relationship type, such as Citibank account or CitiGold, and the customer relationship nickname. The customer relationship component 113 has a list of accounts/products associated with a customer, a list of account categories, and a list of accounts for the customer. The customer relationship component 113 also knows the customer's predefined transactions and has an account summary status. The customer relationship component 113 has the list of payees and the payee list status. The customer relationship component 113 knows the customer privileges or limitations, such as the number of quotes allowed for that customer. Some businesses, such as those in Mexico, Venezuela, or Brazil, can have multiple relationships per card. In the top level navigation, the customer may select one of them as the primary relationship to use for a session. The transfer application, however, can transfer between accounts in different relationships.

Detailed Description Text (50):

The account component 115 contains information and can answer questions about a particular account. Each individual account preferably has only one account component 115 with the account details and rules varying for the particular individual account. The account component 115 supports query of account information and supports update of account information. The account component 115 knows the business owning the account, the category of the account, and the product type and subproduct type of the account. The account component 115 also knows the fund family code and fund code, the category code, the account name, account number, and account details, such as currency code, balances, and terms. The account component 115 has information on the functional privileges and limitations and also information on associated link accounts. The individual accounts may be customer owned or payee accounts that can be the target of a transfer or bill payment.

Detailed Description Text (54):

The language man component 122 provides the application with a facility to resolve the necessary text phrase needed in a particular context. The context includes the language selected by the customer and the type of device in use. The language man component 122 provides a repository of phrases which allows an application to be written in a language and device independent way. From the application point of view, all phrases are named. When an application needs to display a phrase, the application queries the language man component 122 for the correct text for this phrase name given a specified language choice and the current presentation device type, which has been provided by the presentation manager component 52. The language man component 122 can also extend this capability to the use of phrases with imbedded variables. Thus, the application may supply additional parameters to be inserted into the phrase at a required point. To resolve a request, the language man component 122 uses a phrase repository to look up the correct version of a particular phrase, with the repository being segmented. A set of "global" phrases are usable by all applications and a mini-app dialog specific set of phrases is established. Thus, given the ID of a requesting mini-app dialog component 83, the repository specific to that mini-app dialog component 83 is searched first and then, if the phrase is not found, a global repository is searched. The phrase repository allows a degree of independence in the creation of mini-app dialog components 83. No coordinated update to the global repository is needed to release a new mini-app dialog component 83 and a mini-app dialog 83 can override the global phrase. The language man component 122 also provides APIs for the dynamic construction of phrases needed to deal with gender and plural issues encountered in some languages. The language man component 122 looks up a requested phrase in a phrase repository and returns the phrase based on the client ID, language ID, locale, phrase medium, phrase formulate, and device type and may be qualified by the device, as well, such as short form of the phrase for a small display on the

device. The language man component 122 is backed by a set of development tools to create and maintain phrase repositories. These development tools provide for creation and deletion of phrase IDs, mechanisms to add, change, and delete phrase text in the repository, multi-lingual text entry, and specification of variable insertion points as well as graphic files or sound or video files.

Detailed Description Text (56):

The session services set 130 includes a session controller component 131, a session component 132, and a delivery capabilities component 133. The session controller component 131 manages all the sessions in the delivery system 12. When a new customer contacts the delivery system 12, the session controller component 131 starts a session by instantiating a session bubble for the session. The session bubble, for instance session bubble S shown in FIG. 2, bounds a secure set of resources allocated to one and only one customer session. The session controller component 131 is aware of the type of customer remote device a start session request came from and the broad product type of service requested so that the appropriate type of session bubble can be instantiated. The session controller component 131 creates a session when a customer contacts the delivery system 12 by instantiating a new instance of the session object. The session controller component 131 maintains a registry of all active sessions with handles to the session objects. The session controller component 131 also terminates a session when a customer abnormally breaks the connection.

Detailed Description Text (61):

An example session will now be described with reference to FIGS. 3A to 3C and FIGS. 4A to 4C. At a step E1, a customer initiates a session. The customer may initiate a session in various ways depending upon the remote device used to communicate with the delivery system 12. For instance, the customer may use a screen phone 14, a CAT/CASST 16, a personal computer 18, or a PDA 20. The customer may also use a remote device 24 and an external service provider 22 to communicate with the delivery system 12. The customer, regardless of the particular remote device used, initiates the session through the touch point and display component 31 of the delivery system 12. At a step E2, a start banking message is sent from the touch point and display component 31 to the touch point interface component 41. At step E3, the touch point interface component 41 sends the start session message to the session controller component 131. At step E4, the session controller instantiates session component 132. At step E5, the session component 132 then instantiates the delivery capabilities component 133 and the session device manager component 63.

Detailed Description Text (63):

At step E12, the welcome mat component 81 sends a logon to presentation manager component 52. The presentation manager component 52, at step E13, formats the screen based on device specific template and sends formatted information to the front door man component 51. At step E14, the front door man component 51 assigns a session cookie and sends a response via the touch point interface component 41 to the customer.

Detailed Description Text (64):

As reflected in steps E1 through E14, a customer can access the delivery system 12 with any type of remote device. In response, the delivery system 12 will create a session bubble specific for that customer. This session bubble will preferably have a session component 132, a delivery capabilities component 133, a session device manager component 63, a rule broker component 121, a welcome mat component 81, a front door man component 51, as well as various other components dedicated for that particular session. Through the presentation manager component 52, front door man component 51, touch point interface component 41 and touch point and display component 31, the delivery system 12 can format messages to any type of remote device and can custom tailor this message according to the desires of a particular customer. The delivery system 12 is also capable of providing uniformity across the various remote devices so that the customer is presented with a consistent and

familiar interface regardless of the remote device used.

Detailed Description Text (65):

B. Customer Authentication

Detailed Description Text (66):

An example of the process of authenticating a customer will now be described with reference to FIGS. 5A to 5D and FIGS. 6A to 6C. At step E21, a customer enters his or her CIN and PIN at the touch point and display component 31 which forwards the information to the touch point interface component 41. At step E22, the touch point interface component 41 forwards the message to the appropriate session bubble based on the session ID in the session cookie. At step E3, the front door man component 51 performs a security check on the cookie and other parameters before forwarding the message to the presentation manager component 52. At step E24, the presentation manager component 52 routes the input to the dialog services set 80. For instance, the presentation manager component 52 may route the input based on mime type and URL to the appropriate dialog welcome mat component 81.

Detailed Description Text (67):

At step E25, the welcome mat component 81 asks the rule broker component 121 who is the issuer based on the CIN. The welcome mat component 81, in turn, instantiates the customer ID component 111 at step E26 and instantiates the issuer component 112 at step E27. At step E28, the welcome mat component 81 instantiates the profile transaction executor component 91 for authenticating the customer and then passes the CIN and encrypted PIN to the transaction executor component 91. At step E29, the transaction executor component 91 formats a reply message and sends the message to the host through the back door man component 101. At step E30, the back door man component 101 adds a universal message sequence and, at step E31, the external service provider interface component 102 provides protocol gateway to the external service provider 22.

Detailed Description Text (68):

At step E32, a response is returned to the back door man component 101 and the back door man component 101 routes the response to the appropriate transaction executor component 91. At step E33, the transaction executor component 91 extracts information from the external service provider message and gives this information to the welcome mat component 81. At step E34, the transaction executor component 91 instantiates the customer relationship component 113 which, in turn, instantiates the account components 115 at step E35. At step E36, the welcome mat component 91 instantiates the navigation shell component 82 which sends initial navigation choices to the customer at step E37. At step E38, the presentation manager component 52 formats style of screen display and sends a response to the customer via the front door man component 51, touch point interface component 41, and touch point and display unit 31.

Detailed Description Text (70):

The selection of a mini-app will now be described with reference to FIGS. 7A and 7B and FIGS. 8A and 8B. At step E41, the customer selects a mini-app with the touch point and display component 31 and the request is sent into the delivery system 12. At step E42, the presentation manager component 52 demultiplexes the request based on mime-type and URL and sends the request to the navigation shell component 82. At step E43, the navigation shell component 82 instantiates the appropriate mini-app dialog component 83. At step E44, the mini-app dialog component 83 returns choices to the customer. At step E45, a back and forth dialog occurs between the customer and the mini-app dialog component 83 until all information is collected for a function. During this step, the mini-app dialog component 83 directs business rule questions to the rule broker component 121 for resolution during the dialog.

Detailed Description Text (71):

At step E46, after all information has been collected, the mini-app dialog

component 83 instantiates the transaction executor component 91 for the selected function. At step E47, the transaction executor component 91 formats a message to the external service provider 22 and does the transaction with the external service provider 22. At step E48, the transaction executor component 91 extracts information from the host message and passes the information to the mini-app dialog component 83. As step E49, the mini-app dialog component 83 formulates content of response and sends the response to the presentation manager component 52 for formatting. At step E50, the presentation manager component 52 formats style and layout of response and sends the response to the customer via the front door man component 51, touch point interface component 41, and touch point and display component 31.

Detailed Description Text (86):

An example of a NetCAT server 200 is shown in FIG. 9. The NetCAT server 200 has the ability to present a traveling customer their "home screens." This ability is accomplished without the need to load CAT software for all regions on all CATS 16 around the world. The basic notion is to have at least one NetCAT server 200 for every region. On this NetCAT server 200, a region's CAT software will run and it will be capable of being "remotely projected" through any acquiring CAT 16 around the world, thus providing almost all of the customers home screens around the world. Differences from the customer's home screen will show up on the initial welcome screen, until the customer's issuer is identified, and during certain transactions, notably cash withdrawal, where foreign exchange rates will have to be displayed and regulatory requirements of the acquiring country will have to be honored.

Detailed Description Text (87):

To start a NetCAT session, the traveling customer dips his or her card at "foreign" CAT 16 and a session bubble starts up normally at the CAT 16. When the welcome mat component 81 determines that this customer is off-region, the welcome mat component 81 makes a connection to the appropriate regional NetCAT server 200. The welcome mat component 81 on the CAT 16 communicates with the session controller component 131 on the NetCAT server 200 to start up a session. The welcome mat component 81 on the NetCAT server 200, after given card parameters upon start up, instantiates the customer ID component 111 and issuer components 112 on the NetCAT server 200. After NetCAT server 200 authenticates the customer, with its own external service provider, the NetCAT server 200 starts up a navigation shell component 82 on the NetCAT server 200. The CAT 16 exposes/copies certain of its components to the NetCAT server 200 for its use. The CAT 16, for instance, exposes the session component 132, the acquirer component 114, the delivery capabilities component 133, the front door man component 51, the peripheral device manager component 62, the transaction executor components 91, and the welcome mat component 81. The NetCAT server 200 uses these components for business rule inquiries, for delivery to CAT screen, for operation of the CAT peripherals, and for inquiry about the capabilities of the hosting CAT, such as fonts supported and pictographic printing. An example of a CAT 16 is shown in FIG. 10 with the exposed components marked with a black dot.

Detailed Description Text (89):

The delivery system 12 supports an orderly migration of CAT functionality from implementation with AGS applications to implementation with service components on all platforms where AGS is used to delivery CAT look-and-feel functionality. An example of the interaction between CAT AGS applications and service components will be described with reference to FIG. 11. The AGS applications are executed within an instance of a TAFE process, the legacy run time AGS driver and associated functionality, and share a single persistent global data store. At the time a CAT application is invoked, session context is completely represented by the current state of the persistent global data and the content of the Exit message TAFE passes to the application. If this context can be instantiated by alternate means, then the business/customer functionality normally performed by the AGS level one and

level two applications need not be performed before running a level three transaction application. At a high level, pretransaction session context is imported to the TAFE and a level three application is invoked with Exit message. After a return from level three application with Exit message, post transaction session context is exported from TAFE. For the case of a complete session performed in AGS, the interaction includes importing prelanguage selection session context to TAFE, invoking level two application with Exit message, returning from level two application with Exit message, and exporting post end-of-session context from TAFE.

Detailed Description Text (91):

The delivery system 12 is not limited to any particular manner for initiating a CAT AGS application. As an example, however, a pre-initialized TAFE AGS driver process is associated with the session bubble. Within the bubble, a faceless level one application waits on receipt of a start of session context message. The legacy app bridge component 84 for the customer selected transaction sends a start of session context message to the TAFE including track two data. The message sent to the TAFE preferably does not contain data from an element ID range specific to a card issuer. The level one application receives the message and updates session context and persistent global memory. Using the track two data, preinitialized static tables, and existing functionality, the level one application creates and sends the exit message to invoke the level two application appropriate to the card issuer. In this example, the level two application is a faceless, special purpose replacement for the original level two application. The level two application is specific to the element ID range of the issuer and sends a request message for the remainder of the session context data. The request message is routed from the TAFE to the legacy app bridge component 84.

Detailed Description Text (92):

The legacy app bridge component 84 queries other service components in order to construct and return a response message containing the remainder of the session context, including data in the element ID range specific to the level two application that sent the request. The level two application receives the message and updates the session context and persistent global memory. Using the transaction type code, language code, and application state code received in the context data, together with existing functionality, the level two application creates and sends the exit message to invoke the level three application appropriate to the transaction type. The level three application processes the transaction and presents screens, sends and receives external service provider messages, device messages and logging messages, and updates session context in persistent global memory. Upon completion, the level three application sends an Exit message to return to the level two application. The level two application sends a message containing the updated post transaction session context which TAFE routes to the legacy app bridge component 84. The level two application also sends an Exit message to return to the level one application. The level one application waits in receipt of another start of session context message. The legacy app bridge 84 receives the post transaction session context and processes it causing the session context to be updated in the other appropriate service components. In this example, the level one and level two applications perform no customer or business functionality. The role of the level one and level two applications instead is preferably limited to receiving and returning context data and invoking the appropriate lower level application. The delivery system 12, however, can vary from that described above.

Detailed Description Text (103):

A set of complex rules, spanning multiple configuration tables, is used in the AGS implementation when choosing what dispense amounts are displayed on selection buttons to a customer withdrawing cash at a CAT 16. The existing "withdraw cash" application is tightly coupled to the structure of these tables. The acquirer component 114 might register as a rule authority for the "WhatDispenseAmounts?"

question. The input parameters for this question include the product type, which specifies the product being withdrawn from, and the currency. The output parameters include the result code and the variable length list of amounts. Some of the session data needed to answer the question, such as card type and level of service, is available from known session components and consequently is not passed as input. The acquirer component 114, in processing the request, may query whatever database contains specific rules for dispense amounts and ask the peripheral device manager component 62 to determine what denominations are available.

Detailed Description Text (108):

The delivery system 12 is also not limited to any particular integrated development environment (IDE). The IDE, however, should have support for multi-user shared development and should have integration with a configuration management capability. The IDE should also support a tool "plug-in" capability to allow tools to be added which are unique to the delivery system 12. Some examples of these "plug-in" tools include configuration tools to allow for the maintenance of system configuration information and test tools including host and device emulators. Other tools include software distribution tools to standardize the method by which software upgrades are distributed, system management and logging tools, security protocols, and middle-ware for distributed object support in legacy system interfaces. Further tools include template development tools for both canonical and device specific templates, a rules database editor, services registry maintenance tools, and language man repository editor. The IDE preferably supports all of the selected targeted languages so as to minimize retraining and allows reuse of "plug-in" of tools across development languages. The operating system for the delivery system 12 is preferably Microsoft's Windows NT but may alternatively operate on other operating systems, such as a Macintosh or a UNIX operating system.

Detailed Description Text (109):

A component in the delivery system 12 may comprise any piece of hardware or software that is potentially independently replaceable with the software components being embodied as either executables (.EXEs) or dynamically loaded libraries (.DLLs). Components generally have well-defined interfaces. An application, in contrast, is a set of components that does a specific business function, such as cash withdrawal and may comprise several components. Each application in the delivery system 12 preferably comprises one or more dialog components 83 for handling the user interface, one or more business rule components 121, and one or more transaction executor components 91 for handling the message interface with external service providers 22.

Detailed Description Text (111):

The delivery system 12 advantageously provides a common application base for customer activated applications for all remote devices. Thus, a financial institution need not have a first delivery system for its ATMs, a second delivery system for its staff tellers, a third delivery system for personal computers or PDAs, and a fourth delivery system for external service providers. Instead, home banking devices such as a personal computer 18, a smart phone 14, an Internet browser remote device 24, and a PDA 20 may all access the books of a financial institution through the delivery system 12. In addition, the delivery system 12 may provide financial services to its customers through its CAT/CASST 16 and to its employees through branch and CSR staff platforms 26.

Detailed Description Text (113):

The delivery system 12 provides state of the art user interfaces. The interfaces provided by the delivery system 12 support integration of standard multi-media elements, such as pictures, video, and audio. The interfaces also support customizations needed for specific devices, languages, countries, and other local business needs. The interfaces further support multiple co-existing application navigation paradigms and also support the user working in multiple application components at a single time.

Detailed Description Text (114):

The delivery system 12 substantially improves development and maintenance cycle time. The delivery system 12 uses prefabricated components and templates instead of "from scratch" development. The delivery system 12 may embrace widely accepted industry standards for component interfaces so that off the shelf "plumbing" may connect components and enable plugging-in third party components. The delivery system 12 supports "plug and play" application components that can automatically configure themselves for the environment and automatically insert themselves into top level navigation menus. The delivery system 12 supports high productivity prototyping/development tools for top level navigation definition and user interface design, making use of predefined look-and-feel standards. The delivery system 12 separates different parts of an application so that changes in one part do not affect other parts.

CLAIMS:

1. A system for delivering financial services to a customer through a remote device, comprising:

a mini-app dialog component; and

a transaction executor component;

the mini-app dialog component for receiving a request for a financial service function from the customer through the remote device, for collecting information from the customer through the remote device, and for instantiating the transaction executor component;

wherein the transaction executor component is for use in performing the financial service function that was requested by the customer.

2. The system as set forth in claim 1, further comprising an interface between the mini-app dialog component and the remote device, the interface routing information from the customer and the request from the customer from the remote device to the mini-app dialog component.

4. The system as set forth in claim 3, wherein the interface routes information from remote devices of a plurality of different types, the types including ATM terminals and home banking terminals.

12. The system as set forth in claim 11, wherein the mini-app dialog component validates the customer information using the rule broker component.

14. The system as set forth in claim 13, wherein the presentation manager component is for converting the information from the customer into a tagged canonical representation and for supplying the tagged canonical representation to the mini-app dialog component.

16. The system as set forth in claim 1, further comprising a navigation shell component for notifying the customer of available functions and for instantiating the mini-app dialog component based on a function requested.

17. The system as set forth in claim 1, further comprising a welcome mat component for outputting a welcome page to the remote device of the customer and for collecting customer identity and preference information.

18. The system as set forth in claim 17, wherein the welcome mat component instantiates a customer ID component for holding customer information and an issuer component for answering questions on an issuer business.

19. The system as set forth in claim 17, wherein the welcome mat component instantiates a profile transaction executor to authenticate the customer.

20. The system as set forth in claim 17, wherein the welcome mat component instantiates a navigation shell component, the navigation shell component for notifying the customer of available functions and for instantiating the mini-app dialog component based on a function requested.

22. The system as set forth in claim 1, further comprising a language man component for storing a repository of phrases, the mini-app dialog component querying the language man component for a named phrase and the language man component returning the named phrase to the mini-app dialog component according to a language and content specific for the customer and the remote device.

24. The system as set forth in claim 23, wherein the transaction executor component uses the external service provider in performing the function requested by the customer.

25. The system as set forth in claim 1, further comprising a peripheral device component which is called by the mini-app dialog component in response to the function requested by the customer and which is for interfacing with a peripheral device associated with the remote device to perform at least part of the function requested by the customer.

26. The system as set forth in claim 1, further comprising a session controller component for receiving an initial contact from the customer through the remote device and for instantiating a session component for a session bubble associated with the customer.

28. The system as set forth in claim 26, further comprising a touch point interface component for interfacing with a plurality of customers for a plurality of sessions and for routing the information from each customer to the mini-app dialog component associated with the respective session bubble.

31. The system as set forth in claim 1, wherein the remote device is a personal computer.

35. The system as set forth in claim 1, wherein the remote device is a staff terminal used within a financial institution providing the financial services.

37. The system as set forth in claim 1, further comprising a second mini-app dialog component for receiving a second request for a second function from the customer through the remote device, for collecting information from the customer through the remote device, and for instantiating a second transaction executor component for performing the second function.

38. A delivery system for providing financial services to different types of remote devices that include ATM terminals and home banking terminals, the delivery system comprising:

an interface for communicating with the remote devices, the interface being shared between the different types of remote devices;

a mini-app dialog component for receiving requests for financial service functions from the remote devices and for collecting customer information from the remote devices, the requests and the customer information being routed through the interface; and

a transaction executor component, the transaction executor component being

instantiated by the mini-app dialog component and is for use in performing the financial service functions.

44. The method as set forth in claim 39, wherein receiving the request comprises receiving the request from a personal computer.

47. The method as set forth in claim 39, wherein receiving the request comprises receiving the request from a staff terminal located within a financial institution delivering the financial services.

50. The method as set forth in claim 49, further comprising collecting customer identification information and preference information with the welcome mat component.

51. The method as set forth in claim 49, further comprising instantiating under the welcome mat component a profile transaction executor component, the profile transaction executor component for authenticating a customer.

52. The method as set forth in claim 49, further comprising instantiating under the welcome mat component a customer ID component for holding customer information and an issuer component for answering questions on an issuer business.

53. The method as set forth in claim 39, further comprising instantiating a navigation shell component for notifying a customer of available functions and instantiating the mini-app dialog component in response to a selection of a function associated with the mini-app dialog component.

62. The method as set forth in claim 61, further comprising validating customer information with the rule broker component on behalf of the mini-app dialog component.

64. The method as set forth in claim 39, further comprising instantiating a language man component, storing a repository of phrases in the language man component, receiving queries for a named phrase at the language man component, and returning the named phrase according to a language and content specific for a customer using the remote device.

70. A computer-readable medium storing software for use by a computer in delivering financial services to a remote device, the software being executed by the computer in performing a method of:

receiving a request for a financial service function at a mini-app dialog component from a remote device;

collecting information through the remote device with the mini-app dialog component;

instantiating a transaction executor component; and

performing the financial service function that was requested through the remote device with the transaction executor component.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)